Agilent PS-X30 G10103A

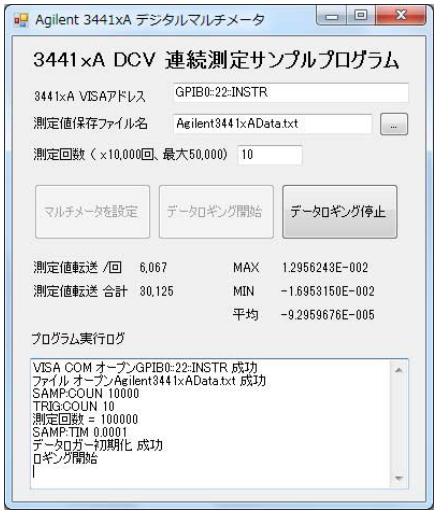
34410A/34411A デジタル・マルチメータ

DCV 連続測定 サンプル･プログラム･ガイド

アジレント社製デジタル・マルチメータ 34410A/34411A/L4411A (3441xA) は、一般的なデジタル・マルチメータの機能に加えて、一定時間間隔で繰り返し測定を行い、結果をメモリに保存することが可能です。さらに、PC から定期的にメモリを読み出すことにより、3441xA のメモリのサイズを超えた回数の連続測定が可能です。

このサンプル･プログラムでは 3441xA を使用して、一定の時間間隔(34410A の最高速度)で、メモリのサイズを超えた回数の連続 DC 電圧測定を行います。

⚠ 大変申し訳ありませんが、弊社では**本サンプル・プログラムの動作保証は行っておりません**。実際のご利用にあたっては、プログラムの内容を十分ご理解いただいた上で、**お客様の責任**で実行してください。



目次

1. サンプル・プログラムの動作環境	P. 2
2. サンプル・プログラムのインストール	P. 2
3. サンプル・プログラムの実行	P. 3
4. サンプル・プログラムの処理内容	P. 4
5. サンプル・プログラム 修正のヒント	P. 6
6. FAQ	P. 8
7. サンプル・プログラムのソース	P. 10
8. 注意事項	P. 15
9. サンプル・プログラムのサポート	P. 15

3441xA Sample Program Guide

*Windows、Visual Basic は、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。

1. サンプル・プログラムの動作環境

◇ ソフトウェア環境

- Windows® XP SP3 以降
- IO Libraries Suite 16.3 （弊社から無償でダウンロード可能）
- Microsoft® Visual Basic® 2010 Express Edition（Microsoft® 社から無償でダウンロード可能）

このサンプル・プログラムは、Visual Basic 2010 Express Edition を使用して作成しています。Visual Basic のコードから 3441xA にコマンドを送り、応答を受け取るため、IO Libraries Suite に含まれる VISA COM ライブラリを呼び出しています。

IO Libraries Suite は、以下の URL よりダウンロード可能です。日本語簡易取扱説明書も合わせてご利用ください。
[IO Libraries Suite 最新バージョン] 2013 年 3 月時点の最新バージョンは 16.3 です。
http://www.agilent.com/find/iolib

[IO Libraries Suite 16.3/16.2 簡易取扱説明書]
http://cp.literature.agilent.com/litweb/pdf/5991-0331JAJP.pdf

◇ ハードウェア環境

上記のソフトウェアが動作する PC
アジレント社デジタル・マルチメータ 34410A、または、34411A、または、L4411A
（GPIB 接続の場合）アジレント社 GPIB インタフェース（82350B、82357B、E5810A など）

2. サンプル・プログラムのインストール

このサンプル・プログラムは ZIP 形式で圧縮されています。適切なフォルダにファイルをダウンロードして、展開してください。

Agt3441xaSample01¥Agt3441xaSample01.sln
プログラムのソリューションファイルです。Visual Basic 2010 Express Edition からこのファイルを指定してプログラムを開いてください。

Agt3441xaSample01¥Agt3441xaSample01¥Agt3441xaSample01.vb
サンプル・プログラムのソースコードが保存されています。

3. サンプル・プログラムの実行

プログラムを実行する PC 上で、Agilent Connection Expert から 3441xA が認識できていることを確認してください。次に、Agilent Connection Expert で、3441xA の VISA Address を確認してください。 Agilent Connection Expert の操作は、IO Libraries Suite 16.3/16.2 簡易取扱説明書をご参照ください。

VISA Address の例 : GPIB0::22::INSTR （GPIB 接続の場合）

: TCPIP0::169.254.4.10::inst0::INSTR （LAN 接続の場合 VXI-11）

: TCPIP0::169.254.4.10::5025::SOCKET （LAN 接続の場合 TCP ソケット）

: USB0::0x0957::0x0607::MY47001234::0::INSTR （USB 接続の場合）

1) Visual Basic 2010 Express Edition を起動します。

「すべてのプログラム」> Microsoft Visual Studio 2010 Express > Microsoft Visual Basic 2010

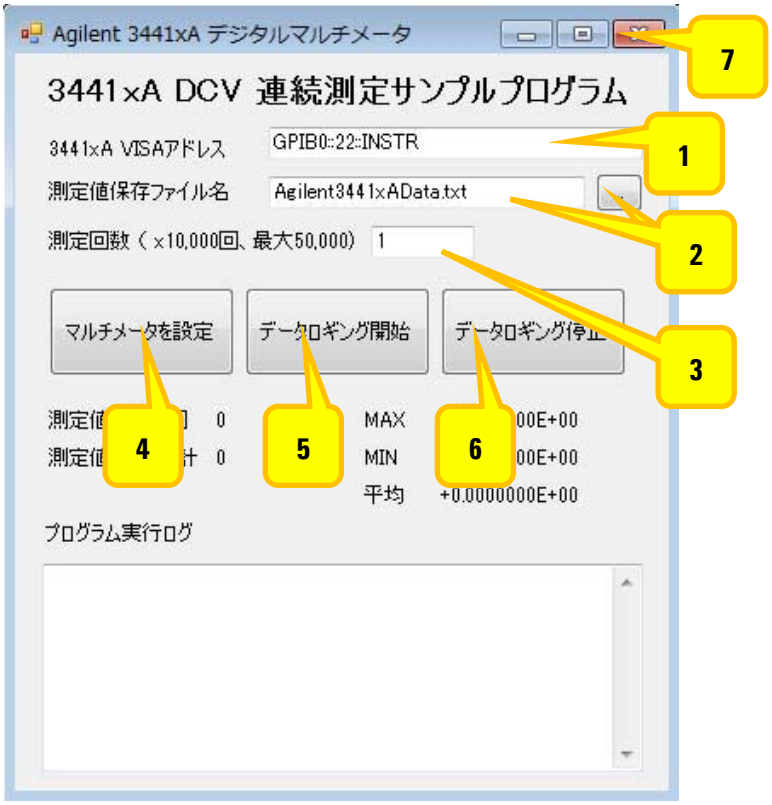
2) サンプル・プログラムのソリューション・ファイルを読み込みます。

ファイル > プロジェクトを開く を実行し、上記 2) で展開した、

Agt3441xaSample01¥Agt3441xaSample01.sln

を選択してください。

3) デバッグ > デバッグ開始 でプログラムが実行されます。



1. VISA アドレスを入力

2. 測定値を保存する
   ファイル名を入力

3. 測定回数を入力
   入力 x 10,000 回測定を
   行います。
   この例では 10,000 回測定

4. 測定のための設定実施

5. 測定開始

6. 測定の中断

7. プログラム終了

4. サンプル･プログラムの処理内容

◇ 3441xA 初期設定部分

このサンプル・プログラムは、3441xA に以下の設定を行います。

測定ファンクション：DC 電圧測定

測定レンジ： 10V レンジ固定

分解能： 最大 (34410A 5･1/2 桁、34411A/L4411A 4･1/2 桁)

積分時間： 最小 (34410A 0.006 NPLC、34411A/L4411A 0.001 NPLC)

トリガ遅延：ゼロ

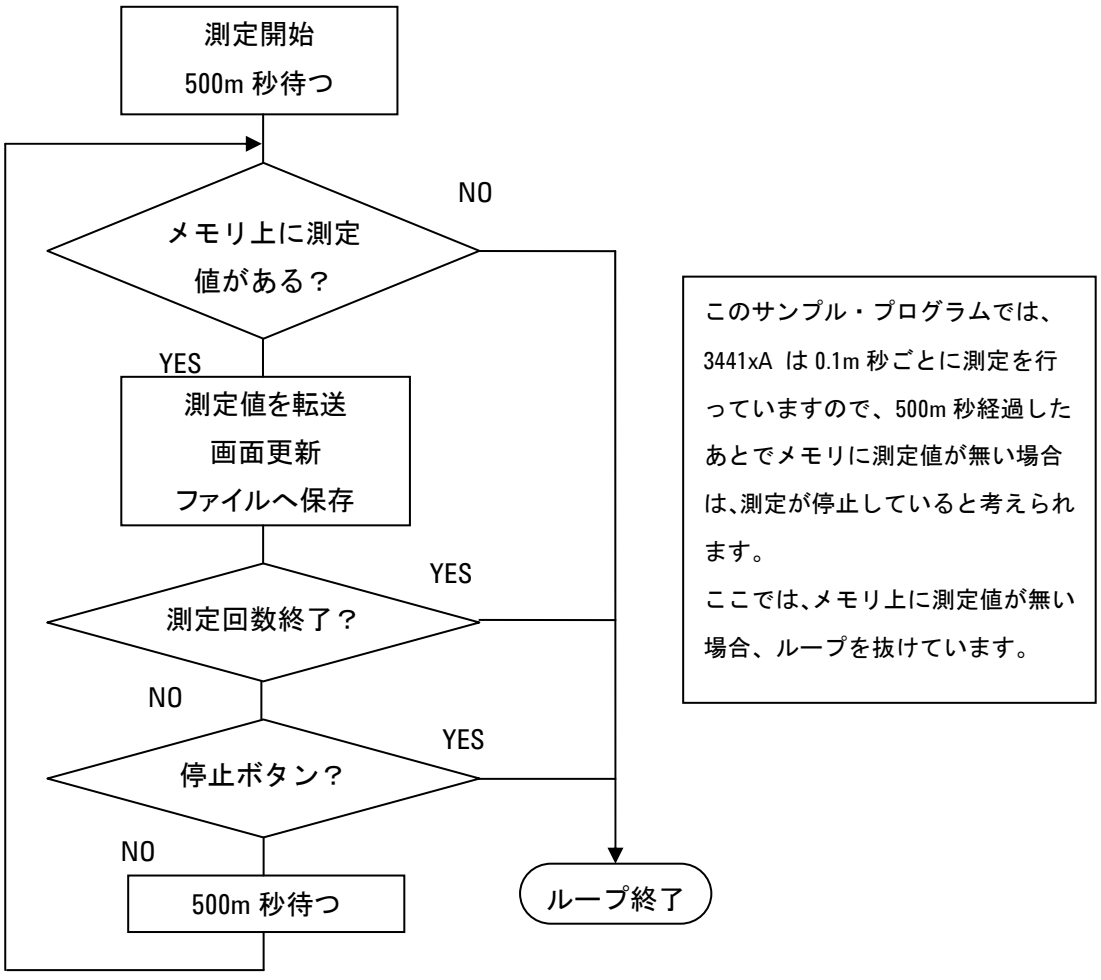
測定間隔： 0.1m 秒

測定回数：10,000 x (画面から入力された数字)

この設定は、34410A で DC 電圧測定を行う場合の最高速度です。34411A/L4411A では、測定間隔を短くすることで、最高で、0.02m 秒の測定間隔の指定が可能です。

◇ 3441xA の測定を開始させ、測定値を転送するループ

初期設定完了後､測定を開始し､500m 秒待ってから測定値転送のループ処理を行います。



◇　測定値転送ループ完了後の処理

測定のループを終了すると、3441xA に、エラーと Questionable Data Register でステイタスを確認します。Questionable Data Register のイベントレジスタを読み出すことにより、3441xA のメモリがオーバーフローしたかどうか、指定した測定間隔で測定が実施できたかどうか、などが確認できます。

メモリがオーバーフローした場合、全ての測定値が PC に転送されていないことを示します。原因は、3441xA が測定を行う速度が、3441xA が測定値を転送する速度より速かったことが考えられます。

指定した測定間隔で測定できなかった場合、設定された測定内容では、指定された測定間隔で測定できないことを示します。3441xA に設定した測定内容と測定間隔を修正してください。

◇　参考資料

3441xA マニュアル

[34410A/11A 6・1/2 桁マルチメータ ユーザーズガイド（日本語版）]

http://cp.literature.agilent.com/litweb/pdf/34410-90413.pdf

[L4411A/34410A/11A Digital Multimeter User's Guide （英語版）]

http://cp.literature.agilent.com/litweb/pdf/34410-90001.pdf

[34410A/11A 6 ½ Digit Multimeter Programmer's Reference （ZIP ファイル）（英語版]

3441xA 製品のページ

http://www.home.agilent.com/agilent/product.jspx?pn=34410a

から、「技術資料 ライブラリ」タブを選択し、「コンテンツのタイプ」を「マニュアル」に指定し、絞り込んでください。

3441xA アプリケーション・ノート

[34410A/11A プログラミング – こんなとき、どんなコマンドを使う？]

http://cp.literature.agilent.com/litweb/pdf/5991-2011JAJP.pdf

VISA COM ヘルプファイル

[IO Libraries Suite ヘルプファイル]

Windows のタスクバー上の「IO」アイコン をクリックし、表示されるメニューから、

Documentation > API Documentation > VISA COM Help

の操作でヘルプファイルが起動します。

5. サンプル・プログラム 修正のヒント

◇　測定の設定

このサンプル・プログラムでは、測定ファンクション：DC 電圧測定、測定レンジ ：10V レンジ固定、測定分解能：最大（34410A:5･1/2 桁、34411A/L4411A: 4・1/2 桁）、積分時間：最小（34410A:0.006NPLC、34411A/L4411A: 0.001NPLC）の設定を行っています。

測定ファンクションを変更する場合、他の測定ファンクションは DC 電圧測定より測定時間が長くなりますので、同時に測定間隔も修正する必要があります。

測定レンジにオートレンジを指定する場合、最適レンジの検出やレンジの変更に時間がかかり、かつ、測定時間が一定ではなくなります。最大の測定時間に合わせて測定間隔も修正する必要があります。

測定分解能を指定すると積分時間が変更されます。積分時間が変更されると測定時間に影響しますので、測定間隔も変更する必要があります。詳細は日本語ユーザーズガイド P.51 をご参照ください。

指定した測定間隔で測定が完了したかどうかは Questionable Data Register で確認できます。

測定関連の設定は、Sub btnSetting_Click() のなかで、以下のコードで実行しています。

```
DMM.WriteString("CONF:VOLT:DC 10, MAX")
DMM.WriteString("VOLT:DC:NPLC MIN")
```

◇　測定間隔

測定間隔は、34410A の最高速度 10,000 回/秒（0.1m 秒/回）の設定を行っています。34411A/L4411A の場合、最高 50,000 回/秒（0.02m 秒/回）まで設定できます。プログラムでは、Sub btnSetting_Click() の中で、

```
intSampTimer = 0.0001
DMM.WriteString("SAMP:SOUR TIM")
DMM.WriteString("SAMP:TIM " & intSampTimer.ToString())
```

で設定しています。

プログラムでは、500m 秒の待ち時間を含むループで、メモリにある測定値の数を調査しています。メモリに測定値がないと、測定を行っていないと判断して測定を終了させています。測定間隔を長くする場合、500m 秒の待ち時間を測定間隔に応じて修正する必要があります。

500m 秒の待ち時間は、Sub btnStart_Click() で、以下のように実現しています。

```
For i As Integer = 1 To 5
        System.Threading.Thread.Sleep(100) ' 100m 秒待つ
        Application.DoEvents()
Next
```

◇　測定回数

3441xA では、（サンプルカウント）x（トリガカウント）の数だけ測定を行います。プログラムでは、サンプルカウントを 10,000 を設定し、トリガカウントは、画面から数値を入力するようにしています。

測定回数を 10,000 以下に修正する場合、プログラムで指定しているサンプル・カウントを修正する必要があります。

◇　タイムアウトエラーが発生する場合

一度に転送する測定値の数によっては、Sub btnStart_Click() の

```
DMM.WriteString (“DATA:REM? ” & intDataPoint.ToString())
sglData = DMM.ReadIEEEBlock()
```

でタイムアウトエラーが発生する場合があります。その場合、Sub btnSetting_Click() 中で

```
‘ DMM.IO.Timeout = 10000
```

のコメントを外してタイムアウトを調整してください。コメントの例では、10 秒です。

◇　TCP ソケットで接続する場合

3441xA を LAN で接続する場合、使用するプロトコルは、VXI-11、TCP ソケットから選択できます。TCP ソケットを選択すると VXI-11 と比較して高速なデータ転送が可能です。TCP ソケットを使用する場合、Sub btnSetting_Click() で記載されている

```
' DMM.IO.TerminationCharacterEnabled = True    ' TCP ソケットで接続する場合
```

のコメントを外してください。IO.TerminationCharacterEnabled = True は、終端文字で読み取りを終了させる設定です。

GPIB や USB、LAN（VXI-11 プロトコル）で接続する場合は、この設定は不要です。

◇　測定値の転送フォーマット

3441xA では、測定値の転送フォーマットとして、ASCII、32 ビット実数、64 ビット実数から選択できます。ここでは、転送速度を優先して 32 ビット実数を使用しています。

転送フォーマットとして、64 ビット実数、もしくは、ASCII を使用すると、有効数字の桁数が増えますが、転送時間が長くなります。

プログラムでは、32 ビット実数の指定を Sub btnSetting_Click() で実施しています。

```
DMM.WriteString("FORMAT:DATA REAL,32")
```

読み取りは、Sub btnStart_Click() で実施しています。

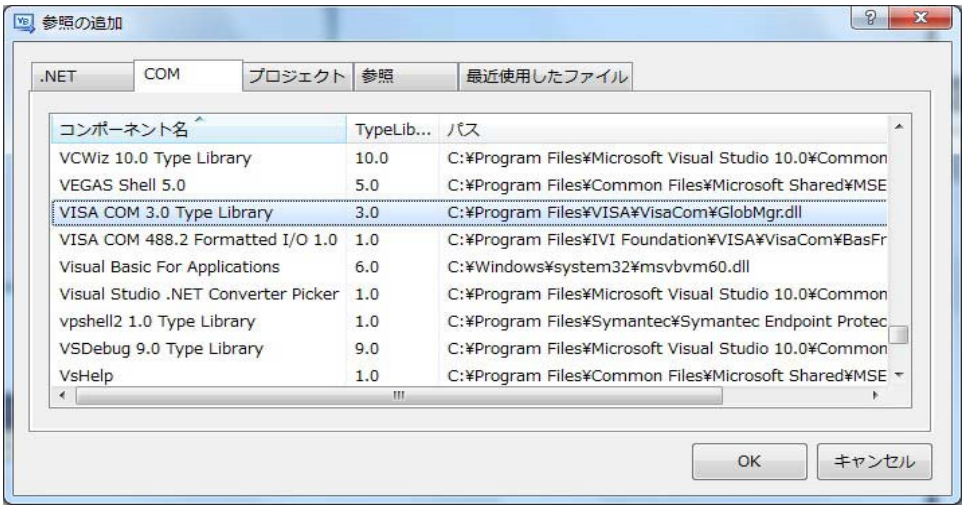
```
DMM.WriteString("DATA:REM? " & intDataPoint.ToString()) ' メモリの測定値を転送
sglMeasData = DMM.ReadIEEEBlock(Ivi.Visa.Interop.IEEEBinaryType.BinaryType_R4)
```

6. FAQ

◇　VISA COM 参照の追加

　このサンプル・プログラムでは 3441xA を制御するために、Agilent IO Libraries Suite に含まれている VISA COM ライブラリを使用しています。

　Visual Basic 2010 Express Edition から VISA COM ライブラリを使用するには プロジェクト > 参照の追加 で以下のダイアログを表示させ、『VISA COM 3.0 Type Library 』を選択し、[OK] を押します。



　このサンプル・プログラムでは既に参照に追加されていますのでこの設定は不要です。新規にプログラムを作成する場合はこの設定を行ってください。

◇　VISA COM ライブラリの使い方

　VISA COM ライブラリを使用すると、プログラムから 3441xA にコマンドを送り、3441xA から応答を受けとることができます。VISA COM ライブラリを使用するには、

1)　リソースマネージャオブジェクトをプログラムにつき 1 個作成する。

2)　Formatted IO オブジェクトと IO リソースオブジェクトを、測定器ごとに作成する。

3)　Formatted IO オブジェクト経由で IO リソースオブジェクトを使用し、測定器にコマンドを送り、応答を受け取る。

4)　測定器の制御が終了したら、1)、2) で作成したオブジェクトを解放する。

を行います。

　サンプル・プログラムでは、以下のように使用しています。

[変数定義とオブジェクトの作成]

```
Dim RM As Ivi.Visa.Interop.ResourceManager ' VISA COM リソースマネージャ
Dim DMM As Ivi.Visa.Interop.FormattedIO488 ' VISA COM フォーマッテッド IO
RM = New Ivi.Visa.Interop.ResourceManager
DMM = New Ivi.Visa.Interop.FormattedIO488
DMM.IO = RM.Open(txtVisaAddress.Text)  ' データロガーをオープン
```

[コマンドの出力と応答の受信]

```
DMM.WriteString(“SYST:ERR?”)  ‘ コマンドの出力
strResp = DMM.ReadString()  ‘ 応答の受信
```

[オブジェクトの解放]

```
DMM.IO.Close()
System.Runtime.InteropServices.Marshal.FinalReleaseComObject(DMM)
System.Runtime.InteropServices.Marshal.FinalReleaseComObject(RM)
```

7. サンプル・プログラムのソースコード

```
'*****************************************************************************
' アジレントテクノロジー 3441xA DCV 連続測定 サンプル・プログラム
'*****************************************************************************
'    PS-X30 G10103A
'    アジレントテクノロジー株式会社
'    電子計測本部 アプリケーションエンジニアリング部
'    テクニカルコンタクトセンター
'    2013/03/07   初版作成
'
'*****************************************************************************
' 本サンプル・プログラムの著作権は、アジレントテクノロジー株式会社が所有しています。
' 本サンプル・プログラムを、お客様は、使用、修正、複製、配布することができます。
' 本サンプル・プログラムを使用したことによって生じた全ての障害・損害・不具合
' (含、サンプル・プログラムの不具合)に関して、弊社及び、弊社の所属するいかなる
' 団体・組織とも、一切の責任を負いません。
' 本サンプル・プログラムは、弊社にて検証の上、提供しておりますが、お客様の
' 環境下での動作は保証しておりません。
'*****************************************************************************
'
Public Class Agt3441xaSample01
    '***************************************
    ' クラスレベルの変数宣言
    '***************************************
    ' リソースの宣言
    Dim RM As Ivi.Visa.Interop.ResourceManager ' VISA COM リソースマネージャ
    Dim DMM As Ivi.Visa.Interop.FormattedIO488 ' VISA COM フォーマッテッドIO
    Dim fWriter As System.IO.StreamWriter ' 測定値 File 書き込み用オブジェクト
    Dim intMeasCount As Integer   ' 設定された測定回数
    ' 繰り返し測定中に中断ボタンが押されたかどうかを監視するための変数。
    ' 停止ボタンの処理で、0 から 1 に変更する。初期値は 0。1で繰り返しを中断する。
    Dim stopFlag As Integer

    Private Sub Agt3441xaSample01_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
        '***************************************
        ' プログラム起動時の処理
        '***************************************
        btnSetting.Enabled = True   ' 設定ボタンは押せる。
        btnStart.Enabled = False   ' 開始ボタンは押せない。
        btnStop.Enabled = False   ' 停止ボタンは押せない。
        stopFlag = 0   ' 停止ボタンは押されていない。
    End Sub

    Private Sub btnSaveFileDialog_Click(sender As System.Object, e As System.EventArgs) Handles btnSaveFileDialog.Click
        '***************************************
        ' ファイル名選択ダイアログボックスボタンが押された時の処理
        '***************************************
        ' ... ボタンの処理
        ' ファイル名を選択するダイアログを表示し、ファイル名を入力する。
        ' 選択されたファイル名は、txtSaveFileName.Text に保存する。
        Dim FileDialog As New SaveFileDialog()

        FileDialog.FileName = "Agilent3441xAData.txt"   ' デフォルトのファイル名
        FileDialog.Filter = "text file(*.txt)|*.txt|csv file(*.csv)|*.csv|all file(*.*)|*.*" 'ファイルの種類
        FileDialog.Title = "測定値を保存するファイルを指定してください。"
        FileDialog.OverwritePrompt = False   ' 上書きの警告は表示しない。

        If FileDialog.ShowDialog() = DialogResult.OK Then   ' ダイアログを表示し、
            txtSaveFileName.Text = FileDialog.FileName   ' テキストボックスにファイル名を保存
        End If
```

```
End Sub

Private Sub btnSetting_Click(sender As System.Object, e As System.EventArgs) Handles btnSetting.Click
    '****************************************
    ' 設定ボタンが押された時の処理
    '****************************************
    ' VISA リソースの作成
    ' VISA リソースの作成で例外が発生すると処理を抜ける
    Try
        RM = New Ivi.Visa.Interop.ResourceManager
        DMM = New Ivi.Visa.Interop.FormattedIO488
        DMM.IO = RM.Open(txtVisaAddress.Text)  ' 3441xAをオープン
        ' DMM.IO.Timeout = 10000  ' VISA タイムアウト 10秒に設定
        ' DMM.IO.TerminationCharacterEnabled = True  ' TCPソケットで接続する場合
    Catch ex As Exception
        addMessage("ERROR: VISA COM オブジェクト作成失敗。 " & ex.Message)
        MsgBox("VISA COMオブジェクトの作成が失敗しました。" & ex.Message)
        Exit Sub
    End Try
    addMessage("VISA COM オープン" & txtVisaAddress.Text & " 成功")
    ' ファイルリソースの作成
    ' ファイルの作成で例外が発生すると処理を抜ける
    Try
        fWriter = New System.IO.StreamWriter(txtSaveFileName.Text, True) ' 追記モード
    Catch ex As Exception
        addMessage("ERROR: ファイル オープン" & txtSaveFileName.Text & " 失敗")
        MsgBox("ファイル名を確認してください。")
        ' ここではVISA リソースはオープンされているのでVISA リソースをクローズ
        DMM.IO.Close()
        System.Runtime.InteropServices.Marshal.FinalReleaseComObject(DMM)
        System.Runtime.InteropServices.Marshal.FinalReleaseComObject(RM)
        Exit Sub
    End Try
    addMessage("ファイル オープン" & txtSaveFileName.Text & " 成功")
    Try
        ' 3441xA 初期化
        Dim intTrigCount As Integer  ' トリガカウント用の変数
        Dim intSampCount As Integer  ' サンプルカウント用の変数
        Dim strResp As String  ' データロガーからの応答を受け取る変数
        DMM.WriteString("*RST;*CLS")  ' リセット、クリアステイタス
        DMM.WriteString("CONF:VOLT:DC 10,MAX")  'DCV、10V レンジに固定、分解能最大
        DMM.WriteString("VOLT:DC:NPLC MIN") ' 積分時間 最少 34410A 0.006 NPLC、34411A 0.001 NPLC
        DMM.WriteString("TRIG:DEL 0") ' トリガ遅延 ゼロ
        DMM.WriteString("FORMAT:DATA REAL,32")  ' バイナリデータ転送 32ビット実数
        ' サンプルカウント
        intSampCount = 10000  ' このサンプルプログラムでは、10,000 に固定
        DMM.WriteString("SAMP:COUN " & intSampCount.ToString())
        addMessage("SAMP:COUN " & intSampCount.ToString())
        ' トリガカウント
        intTrigCount = Integer.Parse(txtTrigCount.Text)  ' 指定されたトリガカウント
        DMM.WriteString("TRIG:COUN " & intTrigCount.ToString())
        addMessage("TRIG:COUN " & intTrigCount.ToString())
        ' トータルの測定回数 = サンプルカウント x トリガカウント
        intMeasCount = intSampCount * intTrigCount  ' 測定回数の計算
        addMessage("測定回数 = " & intMeasCount.ToString())
        ' サンプルタイマ
        Dim dblSampTimer As Double
        dblSampTimer = 0.0001  ' 34410A の最高速度は0.1m秒。 34411A の最高測定速度は0.02m秒。
        DMM.WriteString("SAMP:SOUR TIM")
        DMM.WriteString("SAMP:TIM " & dblSampTimer.ToString())
        addMessage("SAMP:TIM " & dblSampTimer.ToString())
        ' 初期化終了後のエラー有無の確認
        DMM.WriteString("SYST:ERR?")
```

```
            strResp = DMM.ReadString()
            If Integer.Parse(strResp.Substring(0, strResp.IndexOf(","))) <> 0 Then   ' エラーが発生したら
                addMessage("ERROR: データロガー初期化エラー :" & strResp)
                ' リソースの解放
                DMM.IO.Close()
                System.Runtime.InteropServices.Marshal.FinalReleaseComObject(DMM)
                System.Runtime.InteropServices.Marshal.FinalReleaseComObject(RM)
                fWriter.Close()
                Exit Sub   ' 設定ボタンの処理を抜ける。
            End If
        Catch Ex As Exception
            fWriter.Close()
            DMM.IO.Close()
            System.Runtime.InteropServices.Marshal.FinalReleaseComObject(DMM)
            System.Runtime.InteropServices.Marshal.FinalReleaseComObject(RM)
            addMessage("ERROR: 3441xA 設定失敗。" & Ex.Message)
            MsgBox("3441xA の設定が失敗しました。" & Ex.Message)
            Exit Sub
        End Try
        ' データロガー初期化成功
        btnSetting.Enabled = False   ' 設定ボタンは押せない。
        btnStart.Enabled = True   ' 繰り返しスキャンボタンは押せる。
        btnStop.Enabled = False   ' 中断ボタンは押せない。
        addMessage("データロガー初期化 成功")
    End Sub

    Private Sub btnStart_Click(sender As System.Object, e As System.EventArgs) Handles btnStart.Click
        '****************************************
        ' 開始ボタンが押された時の処理
        '****************************************
        Dim intActualMeas As Integer   ' 測定済みの測定数
        Dim intDataPoint As Integer   ' 3441xA 内部メモリのデータ数
        Dim sglMeasData() As Single   ' 3441xA の測定値
        Dim dblMeasSum As Double   ' 平均値計算用の測定値の合計
        Dim dblMeasMax As Double   ' 最大値
        Dim dblMeasMin As Double   ' 最小値

        Try
            DMM.WriteString("INIT")   ' データロギング開始
            addMessage("ロギング開始")
        Catch ex As Exception
            addMessage("ERROR: INIT が失敗しました。")
            Exit Sub
        End Try
        btnSetting.Enabled = False   ' 設定ボタンは押せない。
        btnStart.Enabled = False   ' 開始ボタンは押せない。
        btnStop.Enabled = True   ' 停止ボタンは押せる。
        stopFlag = 0   ' 中断ボタンは押されていない。
        ' 変数の初期化
        intActualMeas = 0   ' 測定済みの数
        dblMeasSum = 0   ' 測定値の合計 平均計算用
        dblMeasMax = Double.MinValue   ' 最大値
        dblMeasMin = Double.MaxValue   ' 最小値
        ' INIT 後、0.5秒待ち、Do Loop による測定値転送ループを開始する。
        For i As Integer = 1 To 5
            System.Threading.Thread.Sleep(100) ' 100m秒待つ
            Application.DoEvents()
        Next
        Try
            ' 測定値転送のメインループ
            '   1) 3441xA のメモリ上の測定値の数を調べる
            '   2) 3441xA のメモリ上に測定値があれば、測定値を転送する
            '   3) 測定値がなければ、測定が実行されていないと判断し、ループを抜ける
```

```
        '    4) 測定値を転送した後で、画面表示を更新し、ファイルに測定値を保存する
        '    5) 指定された回数の測定が完了していれば、ループを抜ける
        '    6) データロギング停止のボタンが押されていれば、ループを抜ける
        '    7) 500m秒待って、ループを繰り返す
        Do
            ' (1) 3441xA  のメモリ上にある測定値の数を調べる
            DMM.WriteString("DATA:POIN?")  ' 34410Aのメモリ上にある測定値の数
            intDataPoint = DMM.ReadNumber(Ivi.Visa.Interop.IEEEASCIIType.ASCIIType_I4)
            If intDataPoint > 0 Then
                ' (2) 3441xAのメモリに測定値がある場合、測定値を転送する
                DMM.WriteString("DATA:REM? " & intDataPoint.ToString()) ' メモリの測定値を転送
                sglMeasData = DMM.ReadIEEEBlock(Ivi.Visa.Interop.IEEEBinaryType.BinaryType_R4)
            Else
                ' (3) 測定値メモリが空の場合は、測定していないと判断し、ループを抜ける。
                abortDmm("測定値メモリ上に測定値がありません")
                Exit Do
            End If
            ' (4) 画面表示を更新し、ファイルに測定値を保存する
            intActualMeas = intActualMeas + intDataPoint   ' 測定数を更新
            For i As Integer = 0 To (intDataPoint - 1)   ' すべての測定値に対して
                dblMeasSum = dblMeasSum + sglMeasData(i)   ' 合計
                If sglMeasData(i) < dblMeasMin Then    ' 最小値
                    dblMeasMin = sglMeasData(i)
                End If
                If dblMeasMax < sglMeasData(i) Then   ' 最大値
                    dblMeasMax = sglMeasData(i)
                End If
                fWriter.WriteLine("{0:E7}", sglMeasData(i)) 'ファイルへ保存
            Next
            ' 画面表示の更新
            lblDataPoin.Text = intDataPoint.ToString("N0")
            lblTotalCount.Text = intActualMeas.ToString("N0")
            lblAve.Text = (dblMeasSum / intActualMeas).ToString("E7")
            lblMin.Text = dblMeasMin.ToString("E7")
            lblMax.Text = dblMeasMax.ToString("E7")
            fWriter.Flush()   ' ファイルバッファをフラッシュ
            ' (5) 指定された回数の測定が完了していれば、ループを抜ける
            If (intActualMeas >= intMeasCount) Then
                Exit Do
            End If
            ' (6) データロギング停止のボタンが押されていれば、ループを抜ける
            If (stopFlag = 1) Then
                abortDmm("停止ボタン検出")
                Exit Do
            End If
            ' (7) 測定中で、指定回数以下で、停止ボタンが押されていないので
            '       500m秒待って、ループを繰り返す
            For i As Integer = 1 To 5
                System.Threading.Thread.Sleep(100) ' 100m秒待つ
                Application.DoEvents()
            Next
        Loop
    Catch ex As Exception
        ' 測定値転送ループ中のエラー
        addMessage("測定が失敗しました。")
        btnSetting.Enabled = True   ' 設定ボタンは押せる。
        btnStart.Enabled = False   ' 開始ボタンは押せない。
        btnStop.Enabled = False   ' 停止ボタンは押せない。
        stopFlag = 0   ' 停止ボタンは押されていない。
        Exit Sub
    End Try

    btnSetting.Enabled = True   ' 設定ボタンは押せる。
```

```
            btnStart.Enabled = False   ' 開始ボタンは押せない。
            btnStop.Enabled = False   ' 停止ボタンは押せない。
            stopFlag = 0   ' 停止ボタンは押されていない。

            addMessage("ロギング完了")
            DMM.WriteString("STAT:QUES:EVEN?") ' Questionable Data Register の取得
            Dim intReg As Integer
            intReg = DMM.ReadNumber()
            Dim strReg As String
            strReg = intReg.ToString("X")
            addMessage("STAT:QUES:EVEN? = 0x" & strReg)
            If (intReg And &H4) Then
                addMessage("サンプルタイマ違反が検出されました")
            End If
            If (intReg And &H4000) Then
                addMessage("メモリオーバーフローが検出されました")
            End If
            Dim strResp As String
            Do
                DMM.WriteString("SYST:ERR?")   ' エラー有無の確認
                strResp = DMM.ReadString()
                addMessage("SYST:ERR? : " & strResp)
                If Integer.Parse(strResp.Substring(0, strResp.IndexOf(","))) = 0 Then
                    Exit Do
                End If
            Loop
            ' リソースの解放
            fWriter.Close()
            DMM.IO.Close()
            System.Runtime.InteropServices.Marshal.FinalReleaseComObject(DMM)
            System.Runtime.InteropServices.Marshal.FinalReleaseComObject(RM)
        End Sub

        Private Sub addMessage(ByVal str As String)
            '****************************************
            ' メッセージを残すサブルーチン
            '****************************************
            txtMessage.Text = txtMessage.Text & str & vbCrLf
            txtMessage.SelectionStart = txtMessage.TextLength   ' TextBoxオブジェクトの最後の行を表示
            txtMessage.ScrollToCaret()
        End Sub

        Private Sub btnStop_Click(sender As System.Object, e As System.EventArgs) Handles btnStop.Click
            '****************************************
            ' 停止ボタンが押された時の処理
            '****************************************
            ' 繰り返しスキャンの繰り返し終了を指示。
            ' 実際には繰り返しスキャン処理の中でstopFlag を監視してループを抜ける。
            stopFlag = 1
        End Sub

        Private Sub abortDmm(ByVal str As String)
            '****************************************
            ' 3441xA 測定を中断するサブルーチン
            '****************************************
            DMM.WriteString("ABOR")   ' 測定の中断
            DMM.WriteString("*OPC?")   ' 中断の完了待ち
            DMM.ReadString() ' 応答の受信 測定中断の完了を検出
            If str <> "" Then
                addMessage(str)
            End If
        End Sub
    End Class
```

8. 注意事項

本サンプル・プログラム、および、本サンプル・プログラム ガイドの著作権は、アジレント・テクノロジー株式会社が所有しています。
本サンプル・プログラムを、お客様は、使用、修正、複製、配布することができます。
本サンプル・プログラム・ガイドの転載、配布はご遠慮ください。

本サンプル・プログラム、および、本サンプル・プログラム・ガイドを使用したことによって生じた全ての障害・損害・不具合（含、サンプル・プログラムの不具合）に関して、弊社及び、弊社の所属するいかなる団体・組織とも、一切の責任を負いません。

本サンプル・プログラム、および、本サンプル・プログラム・ガイドは、弊社にて検証の上、提供しておりますが、お客様の環境下での動作は保証しておりません。

9. サンプル・プログラムのサポート

弊社、テクニカルコンタクトセンターでは本サンプル・プログラムに関して、お電話や Email などで、インストール、実行方法、プログラム内容などのお問い合わせを承っております。お問い合わせには、ご利用の製品のモデル番号（34410A 等 ）及び具体的なお問い合わせ内容 (サンプル・プログラムでエラーが発生する等) をご連絡ください。エラーが発生している場合、エラーの画面コピーなど、エラーの詳細情報もお送りください。

その他、アジレント社の製品に関して、

- 測定器のコマンドや動作
- 測定器を PC に接続する方法
- IO Libraries Suite による測定器の制御方法

など、ご質問等ございましたらお問い合わせください。

このサンプル・プログラムは PC による測定器の制御方法のご紹介を目的としておりますので、個別のご希望に応じたプログラムの修正などは行いません。また、Visual Studio や EXCEL など、他社製品の使い方やプログラム言語の一般的なご質問に関しましては、一般の書籍または web、Help 等の情報をご確認いただきますようお願いいたします。

この資料に関するお問い合わせ先

アジレント・テクノロジー株式会社 電子計測本部
アプリケーションエンジニアリング部 テクニカルコンタクトセンター
電子計測本部・計測お客様窓口
フリーダイアル ： 0120 - 421 - 345
Email ： contact_japan@agilent.com
[お問い合わせ E メールテンプレート (テキスト)]
http://www.home.agilent.com/upload/cmc_upload/All/otoiawase4.txt

自動計測関連情報

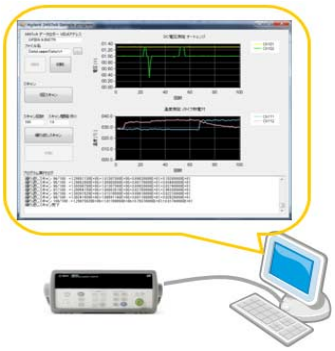
【ソフトウェア】

[IO Libraries Suite 最新版]
IO Libraries Suite は VISA 等を含むコネクティビティ・ソフトウェアです
http://www.agilent.com/find/iolib
2012 年 11 月現在の最新版は 16.3 です。バージョン 16.X は無償でご利用いただけます。

Agilent IO Libraries Suite 16.3/16.2 簡易取扱説明書]
http://cp.literature.agilent.com/litweb/pdf/5991-0331JAJP.pdf

[Agilent VEE Pro 最新版] (計測制御用のプログラム開発環境です)
http://www.agilent.com/find/vee
VEE Pro は有償です。上記より 30 日間の評価版をダウンロード可能です。
2012 年 11 月現在の最新版は 9.3 です。

【サンプル・プログラム（無償）】



サンプル・プログラム、日本語資料のダウンロードが可能です

[Agilent PS-X30 G10101A E3600 DC 電源 サンプルプログラム]
http://www.agilent.co.jp/find/E3600_sample_program

[Agilent PS-X30 G10102A 34970A/34972A データロガー サンプルプログラム]
http://www.agilent.co.jp/find/datalog_sample_program

[InfiniiVision オシロスコープ用サンプル・プログラム ライブラリ]
http://www.agilent.co.jp/find/InfiniiVision-sample

【定期開催トレーニング】

【特長】

ポイント1　受講者の高い満足度

受講者の50%以上が上司/同僚からの推薦。
さらに、受講後アンケートで88%が「他の方に勧めたい」と回答

ポイント2　少人数制。1人1台での測定器実機実習

「実機演習有り」のコースでは、測定器を受講者1人1台使用できますので、自分のペースで実習できます。
「実機演習無し」のコースでも、少人数制ですので、わからない点は講師に質問しやすい環境

ポイント3　お客様先でのトレーニング実施もお勧め

定期開催以外にも、経験豊富な講師が、御社にてトレーニングを実施いたします。新人研修などにご利用ください

ポイント4　受講後のアフターサポートも安心

計測機器ベンダーサポートランキングで、7 年間連続の総合 No1 獲得。
専任エンジニアが常駐し、受講内容・測定方法のアドバイスを提供

[トレーニングコース一覧](日程も記載しています)
http://www.agilent.co.jp/FIND/training

[自動計測トレーニングコース詳細]
http://www.agilent.co.jp/find/programing105
[Excel で始める測定器制御]
[Visual Basic.net による測定器制御 - VISA COM ライブラリ編 -]
[Excel® VBA による測定器自動制御入門 オシロスコープ編]
[VEE 9.3 基礎]
[VEE 9.3 プログラミング]

アジレント・テクノロジー株式会社
本社〒192-8510 東京都八王子市高倉町 9-1

計測お客様窓口
受付時間 9:00-18:00(土・日・祭日を除く)
TEL ■■ 0120-421-345
(042-656-7832)
FAX■■ 0120-421-678
(042-656-7840)
Email contact_japan@agilent.com
電子計測ホームページ
www.agilent.co.jp

記載事項は変更になる場合があります。
ご発注の際にご確認ください。

©Agilent Technologies. Inc. 2013

Published in Japan, March 18,2013
5991-2173JAJP
0000-08A

Agilent Technologies